FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

COMPUTER APPLACATION LAB

ENCS4110

Report#2

EXP.No.8.Timers interrupts

.....

Student Name: saja asfour

Student ID: 1210737

Instructor name: Abdel Salam Sayyad

Section number:2

Date: 3/6/2023

8.1 Abstract:

-The Aim of the experiment: to understand the internals timers/counters and to learn how to setup nested vector interrupt controller (NVIC) , and learn how to configure the timers to make internal interrupt

-Equipment Used in the experiment: keil vision 5 and TM4C123 Tiva C LaunchPad.

Contents:

List of figure:

8.2 Theory

8.2.1 Introduction to Tm4c123 Tiva c launchPad:

*The TM4C123G is a member of the class of high performance 32-bit ARM cortex M4 microcontroller with a bored set of peripherals developed by Texas instruments.

*The Tiva LaunchPad has a built-in processor clock frequency of up to 80MHz with a floating-point unit (FPU).

*The Cortex-M4F processor also supports the tail changing functionality .

*It also includes a nested vector interrupt controller (NVIC).

*The debugging interfaces used is JTAG and SWD (serial wire debugger) for programming and debugging purposes.

*The TM4C123G has a vast variety of application. It host a variety of communication peripherals, which we can be used to connect all sorts of electronics devices , both sensors and actuators for example , IR sensors ,motors.

*The TM4C123G is basically a thumb2 16/32-bit code , which is 26% less memory and 25% faster than pure 32-bit.

*it has a flexible clocking system , and can also access real time clock through hibemation module.

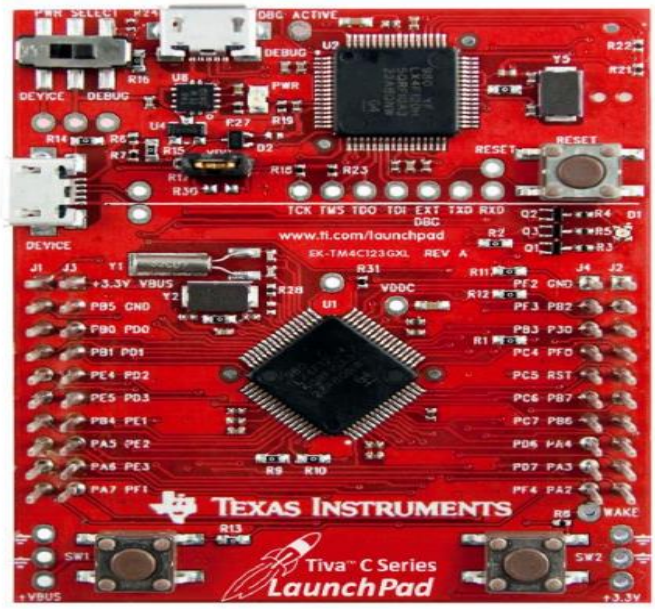

Figure 1: LaunchPad

8.2.2 General Purpose Timer Interrupt:

*Programmable general-purpose timer modules (GPTM) of TM4C123 microcontroller can be Used to count external events as a counter or as a timer.

*For example , we want to measure an analog signal with the ADC of TM4C123 microcontroller after every one second . By using GPTM, we can easily achieve this functionality .

*In order to do this , first configure the timer interrupt of TM4C123 to generate an interrupt after every one second . second, inside the interrupt service routine of the timer, sample the analog signal value with ADC and turn off ADC sampling before returning from the interrupt service routine . similarly, general-purpose timer have many applications in embedded systems.

*TM4C123 microcontroller provides two timer blocks such as timer A and timer B . Each block has six 16/32 bits GPTM and six 32/64 bits GPTM .in this experiment we use TimerA block .

8.2.3 Applications:

-Few important applications are:

- -External event measurement
- -Pulse width measurement
- -Motor RPM measurement TM4C123

8.3 Procedure

8.3.1 TM4c123 Timer Interrupt Example:

In this example , TM4C123 Tiva C Launchpad generated a delay of one second using the Timer1A interrupt handler routine. Inside the main code , we initialized the PF2 pin as a digital output pin.

8.3.1.1 Create a Project:

First, create a project in Keil uvision by selecting TM4C123GH6PM microcontroller

Include the header file of TM4C123GH6PM microcontroller, which contains a .register definition file of all peripherals such as timers.

#Include "TM4C123.h" // Device header file for Tiva Series Microcontroller.

8.3.1.2 Code:

```
/* This is a timer interrupt example code of TM4C123 Tiva C Launchpad */
/* Generates a delay of one second using Timer1A interrupt handler routine */
#include "TM4C123.h" // Device header file for Tiva Series Microcontroller
#define Blue (1<<2) // PF3 pin of TM4C123 Tiva Launchpad, Blue LED
void Timer1A_1sec_delay(void);
int main(void)
{
/*Initialize PF3 as a digital output pin */
 SYSCTL->RCGCGPIO |= 0x20; // turn on bus clock for GPIOF
 GPIOF->DIR |= Blue; // set blue pin as a digital output pin
 GPIOF->DEN |= Blue; // Enable PF2 pin as a digital pin
 Timer1A_1sec_delay();
while(1)
{
// do nothing wait for the interrupt to occur
}
}
/* Timer1 subtimer A interrupt service routine */

TIMER1A_Handler()
{
 if(TIMER1->MIS & 0x1)
GPIOF->DATA ^= Blue; /* toggle Blue LED*/
TIMER1->ICR = 0x1; /* Timer1A timeout flag bit clears*/
}
void Timer1A_1sec_delay(void)
{
SYSCTL->RCGCTIMER |= (1<<1); /*enable clock Timer1 subtimer A in run mode */
TIMER1->CTL = 0; /* disable timer1 output */
TIMER1->CFG = 0x4; /*select 16-bit configuration option */
TIMER1->TAMR = 0x02; /*select periodic down counter mode of timer1 */
TIMER1->TAPR = 250-1; /* TimerA prescaler value */
TIMER1->TAILR = 64000-1 ; /* TimerA counter starting count down value */
TIMER1->ICR = 0x1; /* TimerA timeout flag bit clears*/
TIMER1->IMR |=(1<<0); /*enables TimerA time-out interrupt mask */
TIMER1->CTL |= 0x01; /* Enable TimerA module */
NVIC->ISER[0] |= (1<<21); /*enable IRQ21 */
}
```

Figure 2: Code for example

8.3.1.3 Discussion:

8.3.1.3.1: Timer1A interrupt with one second delay:

We create a timer interrupt of 1Hz . That means the interrupt will occur after every one second because the time period is inverse of the frequency . so whenever an interrupt occurs, we will toggle an LED inside the corresponding interrupt service routine of TimerA module . in short, the interrupt service routine will execute every one second and toggle the onboard LED of TM4C123 Tiva C LaunchPad.

TM4C123 Tiva C LaunchPad has an onboard RGB LED . Blue Led is connected with the PF3 pin of PORTF. We want to toggle this LED inside the interrupt service routine of TimerA .First, we define a symbol name for this PF3 pin by using #define preprocessor directive in line 4.

8.3.1.3.2 : Initialize TimerA registers for one second delay:

In line 28 , we enable the clock to timer block 1 using RCGTIMER register . Setting first bit of RCGTIMER register enables the clock to 16/32 bit general purpose timer module 1 in run mode

In line 29 , disable the Timer1 output by clearing GPTMCTL register.

In line 30, we use a 16/32- bit timer in 16- bits configuration mode . Hence, writing 0x4 to the register selects the 16-bit configuration mode.

In line 31, Timer modules can be used in three modes such as one shot , periodic , and capture and we want the timer interrupt to occur periodically after one second . therefore , we will use the periodic mode . in order to select the periodic mode of timer 1 , we set in line 31 the GPTMTAMR register to 0x02 .

-we are using timer in 16 bit configuration . so the maximum delay a 16-bit timer can generate according to 16MHz operating frequency of TM4C123 microcontroller is given by the equation : 2^16= 65536

16MHz = 16000000

Maximum delay = 65536/16000000 = 4.096 millisecond

Hence, the maximum delay that we can generate with timer1 in 16-bit configuration is 4.096 millisecond .

8.3.1.3.3 : prescaler configuration :

-prescaler adds additional bits to the size of the timer.

-GPTM TimerA prescale (GPTMTAR) register is used to add the required Prescaler value of the timer .

-TimerA in 16-bit has an 8-bit prescaler value .

-prescaler basically scales down the frequency to the timer module . Hence , an 8-bit prescaler can reduce the frequency (16MHz) by 1-255

- we want to generate a one second delay , using a Prescaler value of 250 scales down the operating frequency for timerA block to (16000000/250 =64KHz=64000Hz).

-in line 32, load the prescaler register with a value of 250.

- in line 33, GPTMTAILR register is used to load the starting value of the counter to the timer.

-in line 34, GPTMICR register is used to clear the timeout flag bit of timer

- in line 36, enable the TimerA module , which we disabled earlier.

8.3.1.3.4 : Configure TM4C123 Timer interrupt:

There are two step to enable the interrupt of peripherals in ARM Cortex-M microcontrollers.

-First , enable the interrupt from the Timer module using their interrupt mask register

- in line 35, Bit0 of GPTMIMR enables the TimerA time-out interrupt mask.

- second, we also enable the GPTM interrupt request to nested vectored interrupt controller using their interrupt enable registers.

- interrupt request number of timer1A is 21 or IRQ21. Hence, it corresponds to NVIC interrupt enable register zero.

- in line 27. Enables the Timer1A to interrupt from NVIC level by setting bit21

8.3.1.3.5 : Implementation of Timer interrupt Handler function:

- ISRs (interrupt service routines) of all peripheral interrupts and exception routines are defined inside the startup file of TM4C123 Tiva microcontroller and the interrupt vector table is used to relocate the starting address of each interrupt function

- in order to find the name of the handler function of Timer1 and sub-timer A , by open the start up file we find that the name of Timer1A handler routine is TIMER1A_Handler().

- By using this name of the Timer1A interrupt handler , we write the lines from 20 to 25

- Whenever an interrupt occurs from a particular source , the processor execute a corresponding piece of code (function) or interrupt service routine.

8.3.1.3.6 Steps Executed by ARM Cortex M processor During Interrupt processing:

- Whenever an interrupt occurs , the context switch happens.

-That means the processor moves from thread mode to the handler mode ,

- the ARM Cortex-M microcontroller keeps executing the main application , but when an interrupt occurs , the processor switches to interrupt service routine.

- After executing ISR , it switches back to the main application again .

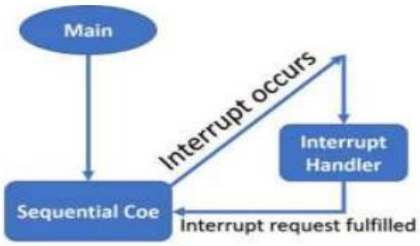

Figure 3: interrupt processing

8.3.1.3.6 ARM Cortex-M Context Switching :

- When an interrupt occurs and its request is approved by the NVIC and processor, the contents of the CPU registers are saved onto the stack.

-This way of CPU registers preservation helps microcontrollers to resume the interrupted program from the same instruction where it is suspended due to an interrupt service routine.

- The process of saving the main application registers content onto the stack is known as context switching.

8.4 Lab Work

8.4.1 Question1:

- Modify the code above to make the GREEN LED blinks every 500ms.

8.4.1.1 Code:

```
/* This is a timer interrupt example code of TM4C123 Tiva C Launchpad */
/* Generates a delay of one second using Timer1A interrupt handler routine */
#include "TM4C123.h" // Device header file for Tiva Series Microcontroller
#define Green (1<<3) // PF3 pin of TM4C123 Tiva Launchpad, Green LED
void Timer1A_1sec_delay(void);
int main(void)
{
/*Initialize PF3 as a digital output pin */
 SYSCTL->RCGCGPIO |= 0x20; // turn on bus clock for GPIOF
 GPIOF->DIR |= Green; // set Green pin as a digital output pin
 GPIOF->DEN |= Green; // Enable PF3 pin as a digital pin
 Timer1A_1sec_delay();
while(1)
{
// do nothing wait for the interrupt to occur
}
}
/* Timer1 subtimer A interrupt service routine */

TIMER1A_Handler()
{
 if(TIMER1->MIS & 0x1)
GPIOF->DATA ^= Green; /* toggle Blue LED*/
TIMER1->ICR = 0x1; /* Timer1A timeout flag bit clears*/
}
void Timer1A_1sec_delay(void)
{
SYSCTL->RCGCTIMER |= (1<<1); /*enable clock Timer1 subtimer A in run mode */
TIMER1->CTL = 0; /* disable timer1 output */
TIMER1->CFG = 0x4; /*select 16-bit configuration option */
TIMER1->TAMR = 0x02; /*select periodic down counter mode of timer1 */
TIMER1->TAPR = 250-1; /* TimerA prescaler value */
TIMER1->TAILR = 32000-1 ; /* TimerA counter starting count down value */
TIMER1->ICR = 0x1; /* TimerA timeout flag bit clears*/
TIMER1->IMR |=(1<<0); /*enables TimerA time-out interrupt mask */
TIMER1->CTL |= 0x01; /* Enable TimerA module */
NVIC->ISER[0] |= (1<<21); /*enable IRQ21 */
}
```

Figure 4: lab task 1

8.4.1.2 discussion:

In this task I modify :

-From Blue led to green led by change #define Blue (1<<3) to #define Green (1<<2) which PF2 pin of TM4C123 Tiva Launchpad , Green Led

- change the TimerA counter starting count down value to 32000 -1 because we want the led blank every 500ms which is 0.5 sec so multiply 64000 by 0.5 to have the new value.

8.4.2 Question2:

- Modify the code above to make the RED LED blinks every 4s.

8.4.2.1 Code:

```
/* This is a timer interrupt example code of TM4C123 Tiva C Launchpad */
/* Generates a delay of one second using Timer1A interrupt handler routine */
#include "TM4C123.h" // Device header file for Tiva Series Microcontroller
#define Red (1<<1) // PF1 pin of TM4C123 Tiva Launchpad, Red LED
void Time1A_1sec_delay(void);
int main(void)
{
/*Initialize PF3 as a digital output pin */
 SYSCTL->RCGCGPIO |= 0x20; // turn on bus clock for GPIOF
 GPIOF->DIR |= Red; // set Green pin as a digital output pin
 GPIOF->DEN |= Red; // Enable PF3 pin as a digital pin
 Time1A_1sec_delay();
while(1)
{
// do nothing wait for the interrupt to occur
}
}
/* Timer1 subtimer A interrupt service routine */

TIMER1A_Handler()
{
 if(TIMER1->MIS & 0x1)
GPIOF->DATA ^= Red; /* toggle Blue LED*/
TIMER1->ICR = 0x1; /* Timer1A timeout flag bit clears*/
}
void Time1A_1sec_delay(void)
{
SYSCTL->RCGCTIMER |= (1<<1); /*enable clock Timer1 subtimer A in run mode */
TIMER1->CTL = 0; /* disable timer1 output */
TIMER1->CFG = 0x00; /*select 32-bit configuration option */
TIMER1->TAMR = 0x02; /*select periodic down counter mode of timer1 */
```

```
TIMER1->TAMR = 0x02; /*select periodic down counter mode of timer1 */
TIMER1->TAPR = 250-1; /* TimerA prescaler value */
TIMER1->TAILR = 64000000-1 ; /* TimerA counter starting count down value */
TIMER1->ICR = 0x1; /* TimerA timeout flag bit clears*/
TIMER1->IMR |=(1<<0); /*enables TimerA time-out interrupt mask */
TIMER1->CTL |= 0x01; /* Enable TimerA module */
NVIC->ISER[0] |= (1<<21); /*enable IRQ21 */
}
```

Figure 5: lab task 2

8.4.2.2 discussion:

In this task I modify :

-From Blue led to Red led by change #define Blue (1<<3) to #define Red (1<<1) which PF1 pin of TM4C123 Tiva Launchpad , Red Led

-modify ti 32 bit

- change the TimerA counter starting count down value to 64000000 -1 because we want the led blank every 4s so multiply 16000000 by 4 to have the new value.

8.4.3 Question3:

- Use the onboard LED and another two externals LEDs with the TM4C123G board to make one LED flashes every 10 seconds , one flashes every 5 sec and one flashes every one second

8.4.3.1 Code:

```
/* This is a timer interrupt example code of TM4C123 Tiva C Launchpad */
/* Generates a delay of one second using Timer1A interrupt handler routine */
#include "TM4C123.h" // Device header file for Tiva Series Microcontroller
#define Red (1<<1) // PF1 pin of TM4C123 Tiva Launchpad, Red LED
#define Yallow (1<<3) // PF3 pin of TM4C123 Tiva Launchpad, yallow LED
#define Green (1<<2) // PF2 pin of TM4C123 Tiva Launchpad, Green LED
void Timer1A_1sec_delay(void);
int main(void)
{
/*Initialize PF3 as a digital output pin */
 SYSCTL->RCGCGPIO |= 0x20; // turn on bus clock for GPIOF
 GPIOF->DIR |= Red |Yallow|Green; // set Green and red and yallow pin as a digital output pin
 GPIOF->DEN |= Red |Yallow|Green; // Enable PF3 and PF1 and PF2 pin as a digital pin
 Timer1A_1sec_delay();
while(1)
{
// do nothing wait for the interrupt to occur
}
}
/* Timer1 subtimer A interrupt service routine */

 TIMER1A_Handler()
{
 if(TIMER1->MIS & 0x1)
GPIOF->DATA ^= Red; /* toggle red LED*/
TIMER1->ICR = 0x1; /* Timer1A timeout flag bit clears*/
}
 TIMER2A_Handler()
{
 if(TIMER2->MIS & 0x1)
GPIOF->DATA ^= Yallow; /* toggle yallow LED*/
TIMER1->ICR = 0x1; /* Timer2A timeout flag bit clears*/
}
 TIMER3A_Handler()
{
 if(TIMER3->MIS & 0x1)
GPIOF->DATA ^= Green; /* toggle green LED*/
TIMER1->ICR = 0x1; /* Timer3A timeout flag bit clears*/
}
 void Timer1A_1sec_delay(void)
{
SYSCTL->RCGCTIMER |= (1<<1); /*enable clock Timer1 subtimer A in run mode */
TIMER1->CTL = 0; /* disable timer1 output */
TIMER1->CFG = 0x00; /*select 32-bit configuration option */
TIMER1->TAMR = 0x02; /*select periodic down counter mode of timer1 */
TIMER1->TAPR = 250-1; /* TimerA prescaler value */
TIMER1->TAILR = 160000000-1 ; /* TimerA counter starting count down value */
TIMER1->ICR = 0x1; /* TimerA timeout flag bit clears*/
TIMER1->IMR |=(1<<0); /*enables TimerA time-out interrupt mask */
TIMER1->CTL |= 0x01; /* Enable TimerA module */
NVIC->ISER[0] |= (1<<21); /*enable IRQ21 */

SYSCTL->RCGCTIMER |= (1<<2); /*enable clock Timer2 subtimer A in run mode */
TIMER2->CTL = 0; /* disable timer1 output */
TIMER2->CFG = 0x00; /*select 32-bit configuration option */
TIMER2->TAMR = 0x02; /*select periodic down counter mode of timer1 */
TIMER2->TAPR = 250-1; /* TimerA prescaler value */
TIMER2->TAILR = 80000000-1 ; /* TimerA counter starting count down value */
TIMER2->ICR = 0x1; /* TimerA timeout flag bit clears*/
TIMER2->IMR |=(1<<0); /*enables TimerA time-out interrupt mask */
TIMER2->CTL |= 0x01; /* Enable TimerA module */
NVIC->ISER[0] |= (1<<23); /*enable IRQ23 */

SYSCTL->RCGCTIMER |= (1<<3); /*enable clock Timer3 subtimer A in run mode */
TIMER3->CTL = 0; /* disable timer1 output */
TIMER3->CFG = 0x00; /*select 32-bit configuration option */
TIMER3->TAMR = 0x02; /*select periodic down counter mode of timer1 */
TIMER3->TAPR = 250-1; /* TimerA prescaler value */
TIMER3->TAILR = 64000-1 ; /* TimerA counter starting count down value */
TIMER3->ICR = 0x1; /* TimerA timeout flag bit clears*/
TIMER3->IMR |=(1<<0); /*enables TimerA time-out interrupt mask */
TIMER3->CTL |= 0x01; /* Enable TimerA module */
NVIC->ISER[0] |= (1<<3); /*enable IRQ3 */
}
```

Figure 6: lab task 3

8.4.3.2 discussion:

In this task I do :

- #define yallow (1<<3) which means I put the yallow led in PF3
- #define Red (1<<1) which means I put red led in PF1
- #define Green (1<<2) which means I put green led in PF2
- in line 12 in main function I set red led and yallow red and green led as a digital output pin
-in line 13 in main function I enable PF3 and PF1 and PF2 pin as a digital pin
-I make handler for each led becouse each of them has a spasific time to work
- in delay function I do for each of them delay like this:

*for red led I want it to work every 10 seconds so I change the counter down value to 160000000-1 which it (10*16000000 -1) and I enable the IRQ21 .

*for yallow red I want it to work every 5 sec so I change the counter down value to 80000000-1 which it (5*16000000-1) and I enable the IRQ23

*for green led I want it to work every one sec and I enable the IRQ3

8.5 Conclusion

In this experiment we discuss TM4C123 Timer interrupt programming on ARM Cortex M4 microcontrollers. First, we discuss application of timer interrupt with one example and use a TM4C123 Tiva C LaunchPad for demonstration purposes . and then we modify the example code in the lab work .

8.6 References

- https://ritaj.birzeit.edu/bzu-msgs/attach/2282152/Exp8_Timer_Interrupt.pdf